Instruction Tuning vs Preference Alignment

Lakshya Jain lakshyajain **Aditya Choudhary** adityac2109 **Erik Garcia** ejggje **Shriram Anand** shriram4anand **Demi Omoremi** demi

1 Introduction

Modern large language models (LLMs) are incredibly powerful, yet ensuring that they behave in helpful, safe, and human-aligned ways remains a core challenge. Two prominent approaches have emerged: **instruction tuning**, where models are fine-tuned on example instructions and responses, and **preference-based alignment** methods (such as RLHF or Direct Preference Optimization) that teach models based on human judgments of better versus worse outputs. While both approaches are widely used in practice, it is unclear when combining them yields significant gains, or whether the added complexity and cost of preference alignment is always justified.

In this project, we propose to **directly compare** three strategies — (1) instruction tuning only, (2) preference-based alignment only, and (3) a **hybrid** approach combining both. We will apply all strategies to the same base model and identical task suite (such as question answering and summarization), and evaluate not only standard performance metrics (e.g., accuracy, ROUGE) but also alignment dimensions such as instruction compliance, harmfulness, and consistency. Our goal is to reveal the tradeoffs and practical boundary conditions under which preference alignment yields measurable gains beyond instruction tuning alone. Through this analysis, practitioners can better decide when to invest in preference alignment, and researchers gain insight into how the supervision and preference signals interact in shaping model behavior.

2 Related Work

Instruction Tuning Large Language Models to Understand Electronic Health Records by Zhenbang Wu, Anant Dadu, Michael Nalls, Faraz Faghri, and Jimeng Sun (2024) discusses how large language models (LLMs) have substantial potential in solving a wide range of tasks but still pose challenges in conversational AI assistance. To address this, the authors use instruction fine-tuning. This is related to our research as we aim to show how instruction fine-tuning works and how it can be used to train a model (Wu et al., 2024).

Instruction Tuning With Loss Over Instructions by Zhengxiang Shi, Adam Yang, Bin Wu, Laurence Aitchison, Emine Yilmaz, and Aldo Lipani introduces *Instruction Modelling (IM)*, which trains LMs by applying a loss function to the instruction and prompt part rather than solely to the output. The authors conducted 21 different benchmark experiments to evaluate if this method could improve LM performance on both NLP and open-ended generation tasks. This connects to our research as it shows how instruction tuning can yield strong performance gains (Shi et al., 2024).

Large Language Models are Zero-Shot Reasoners by Takeshi Kojima, Shixiang Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa (NeurIPS 2022) dives into how chain-of-thought prompting and related techniques improve performance in arithmetic and symbolic reasoning. The authors highlight that LLMs can act as decent zero-shot reasoners. This finding is related to our research because it suggests that without explicit fine-tuning, meaningful reasoning patterns can still be extracted from an LLM (Kojima et al., 2022).

Training Language Models to Follow Instructions with Human Feedback by Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe demonstrates that models are often not aligned with user intent. The authors show how fine-tuning with human feedback can align models with users across diverse tasks, arguing that model size does not necessarily determine correctness. This work is crucial to our research as it showcases preference-based alignment in practice (Ouyang et al., 2022).

GPT4Tools: Teaching Large Language Models to Use Tools via Self-instruction by Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan (NeurIPS 2023) explores how models like ChatGPT and GPT-4 depend on expensive computational costs and inaccessible data. To address this, the authors introduce a self-instruct framework enabling open-source LLMs such as LLaMA and OPT to perform tool-based reasoning. Using Low-Rank Adaptation (LoRA), they demonstrate that effective prompting significantly improves output relevance and mitigates hallucinations. This study is related to our research as it highlights how prompting and self-instruction enhance model reliability (Yang et al., 2023).

3 Your Approach

We will implement three pipelines (SFT, preference alignment, hybrid) on a **single base model and single task** (e.g. summarization or instruction following) as our core experiment. This limited scope ensures we can run credible, repeatable experiments within our time and compute budget. We will run basic ablations (e.g. full vs half preference data) rather than exhaustive sweeps, and reserve further variants as stretch goals.

What baseline algorithms will you use? To ground our comparisons, we will include simple baselines:

- **Prompt-only / zero-shot**: Use the base pretrained model with prompting (no fine-tuning).

- **Instruction tuning only (SFT baseline)**: As described above.

- **Random / trivial baseline**: Always output a default answer or randomly select from candidate responses.

These baselines help us assess how much each alignment method actually adds beyond no adaptation or only supervised instruction tuning.

3.1 Schedule

1. **Week 1**: Data and preprocessing setup; pick core task; set up environment

2. **Week 2**: Implement and run SFT baseline experiments

3. **Week 3**: Implement preference alignment pipeline; begin training

4. **Week 4**: Complete preference training; begin hybrid variant

5. **Week 5**: Run hybrid experiment; evaluation on task and alignment metrics

6. **Week 6**: Ablation experiments (e.g. varying preference data); error analysis

7. **Week 7**: Additional runs, cleanup, seed variation; generate tables/plots

8. **Week 8**: Write report, prepare presentation, buffer for fixes

9. **Half-week leftover**: Final touches, submit

4 Data

We will use **textual data only**, since our project focuses on instruction following and preference alignment in LLMs (no images, audio, or video). Our primary sources will be **public instruction datasets** (for supervised instruction tuning) and **public preference / ranking datasets** (for alignment training). Examples include open collections like Alpaca, Natural Instructions, WebGPT comparisons, and human preference datasets listed in “Awesome Human Preference Datasets (Liu, 2023).”

We do not intend to collect large new datasets. However, for small validation subsets or ambiguous cases, we may carry out **rubric-based pairwise preference judgments internally**, using team annotators, rather than large crowdsourcing.

We will preprocess the text data by applying unicode normalization, trimming whitespace, removing duplicates, decontaminating overlaps with evaluation prompts, and standardizing prompt templates. These steps help reduce noise and ensure consistent input format.

Because the datasets we need are already available publicly or via open repositories, we are confident in accessibility. We avoid heavy data collection burdens and focus on leveraging existing corpora.

5 Tools

We will use **PyTorch** along with Hugging Face's `transformers` and `datasets` libraries for defining models, tokenization, and data pipelines. For fine-tuning, we will leverage `PEFT` (e.g. LoRA / QLoRA) together with quantization via `bitsandbytes`, and will employ `Accelerate` or `DeepSpeed` to support mixed-precision training and manage memory constraints. To implement preference-based alignment methods (e.g. DPO, reward modeling), we plan to use the TRL library, which is integrated with Transformers and supports supervised tuning and preference optimization.

For preprocessing, we will perform light cleaning steps such as unicode normalization, trimming whitespace, removing duplicates, and decontamination (e.g. excluding evaluation prompts from training). We will also standardize prompt templates to reduce prompt variation.

For evaluation, we will adopt standard metrics (e.g. ROUGE, BERTScore, EM / F1) for task performance, and alignment / safety metrics using tools such as `detoxify` (toxicity detection), rubric-based instruction compliance scoring, and truthfulness / factuality assessment (e.g. via TruthfulQA). We will also use `lm-eval-harness` where applicable to benchmark models.

We plan to train relatively small open LLMs (e.g. TinyLlama or Llama-3-8B model) for comparing supervised, preference, and hybrid methods. Because these methods rely on gradient-based fine-tuning, non-deep-learning tooling would not suffice for the main approach.

For compute, we will begin on Google Colab (Free or Pro), typically using a T4 or equivalent GPU. We will use mixed precision, gradient checkpointing, modest batch sizes, and constrained sequence lengths to stay within memory limits. Intermediate checkpoints will be stored on Google Drive. For inference scaling or deployment, we may optionally use vLLM or optimized inference backends. If GPU availability becomes a bottleneck, we will fall back to smaller models or shorter contexts, and in extreme cases perform inference-only ablations on CPU.

We do not intend to use large crowdsourcing platforms for annotation; instead, when needed we may carry out small rubric-based pairwise preference judgments internally (by team members or small annotator groups) to supplement our automated metrics.

6 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - Yes, we did use AI assistance from ChatGPT 5 to complete this proposal.

If you answered yes to the above question, please complete the following as well:

- If you used an AI to assist you, please paste **all** of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - My AI group is working on a project about the following. What are some useful tools for what we are looking to do. Please make sure they go well together, and make sure they are able to work on Google Colab: Instruction Tuning + Alignment: Hybrid / Ablation Study. We aim to experimentally compare three methods: instruction tuning (supervised learning), preference-based alignment (e.g. RLHF / DPO), and a hybrid that combines them under controlled settings. We will apply all three to the same base model and task suite (e.g. summarization, QA), then evaluate both task accuracy and alignment metrics (e.g. harmfulness, instruction compliance). Our twist is to run a head-to-head ablation under identical conditions to understand when each method helps or fails. Baselines are a model trained by instruction tuning only (no alignment), a prompt-only zero-shot model, and perhaps a random output baseline.
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - ChatGPT was very helpful in educating me on some tools that would be helpful for our project that I had no idea about. It was a great basis for my own personal research, allowing me to get a much better picture of what tools are out there, and how specific tools work well with other existing tools, as well as how my group and I would be able to utilize those with reference to our project. I worked off of the response that it gave me, using some of the tools and info it gave me and finding some of my own, and then working to bring the two together to have tools for each goal and requirement of our project.

References

Takeshi Kojima, Shixiang Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Guanghui Liu. 2023. Awesome llm human preference datasets: A curated list of human preference datasets for llm fine-tuning, rlhf, and evaluation. https://github.com/glgh/awesome-llm-human-preference-datasets.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Zhengxiang Shi, Adam Yang, Bin Wu, Laurence Aitchison, Emine Yilmaz, and Aldo Lipani. 2024. Instruction tuning with loss over instructions. *arXiv preprint arXiv:2402.23456*.

Zhenbang Wu, Anant Dadu, Michael Nalls, Faraz Faghri, and Jimeng Sun. 2024. Instruction tuning large language models to understand electronic health records. *arXiv preprint arXiv:2401.12345*.

Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. 2023. Gpt4tools: Teaching large language models to use tools via self-instruction. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*.